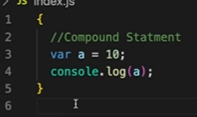
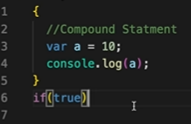
**Block(also known as)**

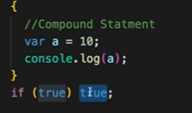
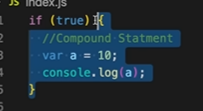
block is defined by {}

we group multiple statements together in a block so that we can use it where js expects 1 statement.



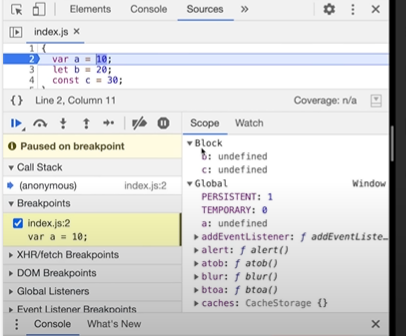
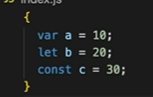


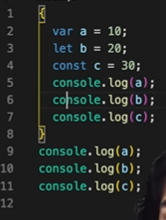
if, expects one statement .

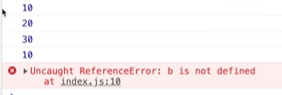


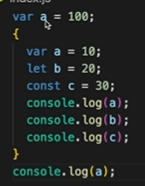


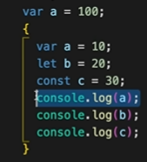
**Scop**:-What all var. & fun() access inside the block.

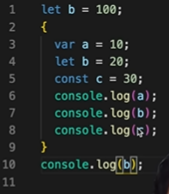


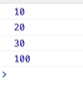
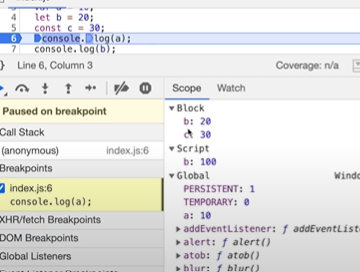


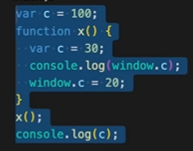


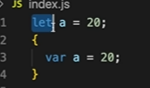
**Shadowing:-**

****

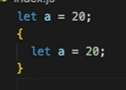
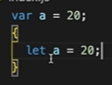
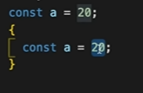




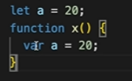
Same in fun()

**Illegal Shadowing:-**

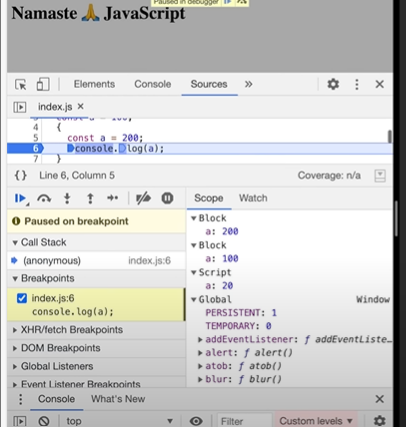
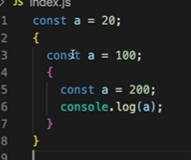


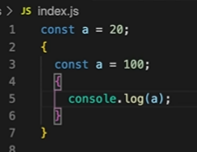




var is fun() scope. Therefore not crosing boindaries.







lx-scope chain

Things learned:

1. Code inside curly bracket is called block.

2. Multiple statements are grouped inside a block so it can be written where JS expects single statements like in if, else, loop, function etc.

3. Block values are stored inside separate memory than global. They are stored in block. (the reason let and const are called block scope)

4. Shadowing of variables using var, let and const.

5. The shadow should not cross the scope of original otherwise it will give error.

6. shadowing let with var is illegal shadowing and gives error.

7. var value is stored in nearest outer function or global scope and hence can be accessed outside block as well whereas same is not the case with let and const.